CLAIMS

What is claimed is:

1. A method comprising:

identifying scratch values generated during speculative execution of a processor; and

setting at least one tag associated with at least one data area of the processor to indicate that the data area holds a scratch value.

2. The method of claim 1 wherein setting comprises setting the tag of a register when an instruction having the register as a destination results in a cache miss.

3. The method of claim 1 further comprising:

bypassing execution of those instructions of a plurality of instructions having at least one operand with an associated tag that indicates that the operand is a scratch value.

4. The method of claim 1 further comprising:

bypassing execution of an arithmetic instruction having as at least one operand a register with an associated tag indicating that the register contains data that is a scratch value; and

bypassing execution of a store instruction involving a value derived from the register having an associated tag indicating that the register contains data that is a scratch value.

5. The method of claim 1 further comprising:

utilizing a branch predictor to override computed branch results produced by branch instructions based on data having a tag that indicates the data is a scratch value.

6. The method of claim 1 further comprising:

marking each instruction in a pipeline with a tag to indicate if the instruction involves a scratch value.

7. The method of claim 1 further comprising:

propagating the tag through a store buffer if an address generation register does not indicate that the address generation register holds a scratch value.

8. A processor comprising:

a plurality of registers having a corresponding plurality of register tags to indicate whether the data stored in the register holds a scratch value;

a plurality of flags having a corresponding plurality of flag tags to indicate whether the data reflected by the flag is based on a scratch value; and

a plurality of predicates having a corresponding plurality of predicate tags to indicate whether the data reflected by the predicate is based on a scratch value.

9. The processor of claim 8 having an instruction set including a plurality of instructions, each instruction augmented by an instruction tag to indicate whether the instruction involves a scratch value.

10. The processor of claim 9 wherein the register tags, flag tags, predicate tags, and instruction tags have a size of one bit.

11. The processor of claim 9 wherein the register tags, flag tags, predicate tags and instruction tags have a size of at least two bits.

12. A processor comprising:

at least two arithmetic units;

a translation look aside buffer;

a branch prediction unit; and

an execution engine having a plurality of instructions which when executed cause the processor to perform actions including:

identifying scratch values generated during speculative execution of a processor, and

setting at least one tag associated with at least one data area of the processor to indicate that the data area holds a scratch value.

13. The processor of claim 12 wherein setting comprises setting the tag of a register when an instruction having the register as a destination results in a cache miss.

14. The processor of claim 12 wherein the execution engine has further instructions which when executed cause the processor to perform further actions comprising:

bypassing execution of those instructions of a plurality of instructions having at least one operand with an associated tag that indicates that the operand is a scratch value.

15. The processor of claim 12 wherein the execution engine has further instructions which when executed cause the processor to perform further actions comprising:

bypassing execution of an arithmetic instruction having as at least one operand a register with an associated tag indicating that the register contains data that is a scratch value; and

bypassing execution of a store instruction involving a value derived from the register having an associated tag indicating that the register contains data that is a scratch value.

16. The processor of claim 12 wherein the execution engine has further instructions which when executed cause the processor to perform further actions comprising:

utilizing a branch predictor to override computed branch results produced by branch instructions based on data having a tag that indicates the data is a scratch value.

17. The processor of claim 12 wherein the execution engine has further instructions which when executed cause the processor to perform further actions comprising:

marking each instruction in a pipeline with a tag to indicate if the instruction involves a scratch value.

18. The processor of claim 12 wherein the execution engine has further instructions which when executed cause the processor to perform further actions comprising:

propagating the tag through a store buffer if an address generation register does not indicate that the address generation register holds a scratch value.

19. A system comprising:

a memory, a storage device, and a processor each coupled to a bus;

the processor including an execution engine having instructions which when executed by the processor cause the processor to perform actions including:

identifying scratch values generated during speculative execution of a processor; and

setting at least one tag associated with at least one data area of the processor to indicate that the data area holds a scratch value.

20. The system of claim 19 wherein setting comprises setting the tag of a register when an instruction having the register as a destination results in a cache miss.

21. The system of claim 19 wherein the execution engine has further instructions which when executed cause the processor to perform further actions comprising:

bypassing execution of those instructions of a plurality of instructions having at least one operand with an associated tag that indicates that the operand is a scratch value.

22. The system of claim 19 wherein the execution engine has further instructions which when executed cause the processor to perform further actions comprising:

bypassing execution of an arithmetic instruction having as at least one operand a register with an associated tag indicating that the register contains data that is a scratch value; and

bypassing execution of a store instruction involving a value derived from the register having an associated tag indicating that the register contains data that is a scratch value.

23. The system of claim 19 wherein the execution engine has further instructions which when executed cause the processor to perform further actions comprising:

utilizing a branch predictor to override computed branch results produced by branch instructions based on data having a tag that indicates the data is a scratch value.

24. The system of claim 19 wherein the execution engine has further instructions which when executed cause the processor to perform further actions comprising:

marking each instruction in a pipeline with a tag to indicate if the instruction involves a scratch value.

25. The system of claim 19 wherein the execution engine has further instructions which when executed cause the processor to perform further actions comprising:

propagating the tag through a store buffer if an address generation register does not indicate that the address generation register holds a scratch value.

26. A machine readable medium having instructions stored thereon which when executed by a processor cause the processor to perform actions including:

identifying scratch values generated during speculative execution of a processor; and

setting at least one tag associated with at least one data area of the processor to indicate that the data area holds a scratch value.

27. The machine readable medium of claim 26 wherein setting comprises setting the tag of a register when an instruction having the register as a destination results in a cache miss.

28. The machine readable medium of claim 26 having further instructions which when executed cause the processor to perform further actions comprising:

bypassing execution of those instructions of a plurality of instructions having at least one operand with an associated tag that indicates that the operand is a scratch value.

29. The machine readable medium of claim 26 having further instructions which when executed cause the processor to perform further actions comprising:

bypassing execution of an arithmetic instruction having as at least one operand a register with an associated tag indicating that the register contains data that is a scratch value; and

bypassing execution of a store instruction involving a value derived from the register having an associated tag indicating that the register contains data that is a scratch value.

30. The machine readable medium of claim 26 having further instructions which when executed cause the processor to perform further actions comprising:

utilizing a branch predictor to override computed branch results produced by branch instructions based on data having a tag that indicates the data is a scratch value.